

# Strategic Web Based Management Game

Group 12

Per Eriksson

Per Strand

Simon Ragnar

Ingemar Markström

Max Walter

# Requirement Document

# **1 Preface**

## ***1.1 Version history***

Version	Comment (reason for / summary of changes)	Date	Author(s)
1.0	First version	2008-01-02	Per Eriksson Per Strand Simon Ragnar Ingemar Markström Max Walter

## ***1.2 Expected readership of the Requirements Document***

The expected readership of this document are software developers and investors.

## 2 Table of contents

1 Preface.....	3
1.1 Version history.....	3
1.2 Expected readership of the Requirements Document.....	3
3 Introduction.....	5
3.1 Who are the users and what problem does the system solve for them? .....	5
3.2 The main uses of the system .....	5
3.3 The context/environment in which the system is to be used.....	6
3.4 The scope of the system.....	6
3.5 The main factors that need to be taken in to account when designing and building the system .....	6
3.6 Technologies and Risks .....	7
4 Glossary.....	10
5 User requirements definition.....	12
5.1 Functional requirement.....	12
5.2 Non-functional requirements.....	16
6 System architecture.....	19
7 System requirements specification.....	20
7.1 Functional requirements.....	21
7.2 Non-functional requirements.....	32
8 Use Cases.....	38
8.1 Web page.....	39
8.2 Game map.....	40
8.3 Resources.....	42
8.4 The ship.....	44
8.5 Modules.....	48
8.6 High-score lists.....	52
8.7 Alliances.....	53
8.8 Communication.....	55
8.9 Research.....	56
9 System evolution.....	58
9.1 Fundamental assumptions on which the system is based:.....	58
9.2 Anticipated future changes to the system.....	58
10 Appendices.....	59
10.1 Database descriptions.....	59
11 Alphabetical Index.....	60

## **3 Introduction**

### ***3.1 Who are the users and what problem does the system solve for them?***

- For this project we are going to produce a web-based, massive multi player, strategy game.
- The main users are expected to be people who like to play web-based massive multi player, strategy and/or role-playing games on PC or console.
- The main purpose of the game is to serve as entertainment over time. Having that said, it's not aimed towards people who are looking for a game to play over several hours in a row but rather as a continual, "in the back of their minds", planning and plotting of strategies to be used in the game.
- The game will also, to a certain degree, provide the users with a social sphere that can become more or less important to the player depending on how long he/she has been playing the game and how much the player likes to interact with other players. For example, for some people the social aspect of the game can be crucial in order to keep them playing.

### ***3.2 The main uses of the system***

The main use of our web-game is for the player to have fun and feel like he/she has accomplished something when he stops playing. The game is set out in a lost dimension in outer space where each player will command his/her space ship. The goal of the game is to escape from this dimension by reaching the middle of the playfield where a wormhole is located. Because of the structure of the game, the time spent on playing the game will be limited to short periods of time.

Players will have a fixed amount of resources which can be spent on various actions. As resources are constantly, but slowly, ticking in for the player, he/she will only be able to make a certain amount of actions at a time. The thought is that the player will be coming back to the game in order to become successful. Because of that, the game will be running continuously on a server where the players' progress will be saved. Players' actions can effect even offline players as the game will be constantly updated.

The player will be able to come back to the game universe and continue on from where he/she left off, but the players around him will probably have made some moves too, and even made moves affecting him. You can interact with other players through the forum which will be up on the web-site and by sending private messages to each other. The main objective of the game is not necessarily to steal all of the player's concentration, but to give the him/her the possibility to play the game along with other games.

#### Example of usage:

Arnold, a 15 year old student, is waiting for an online multiplayer game to start. He glances at his watch, and realizes that he has the possibility to log in and make a move in our game. He opens up a web-browser, goes to the game-page, and comes to the login screen. After logging in, a new screen appears where he can either choose to manage his account or play the game. He chooses to play the game and clicks on button which views the game map. There he can see where he is on the play field, nearby enemies, friends, teammates and other relevant information in lists. The player decides

to move his ship closer to a teammate and shoots an enemy with his missiles. He then realizes that he doesn't have any resources to make any more actions so he quits the game by logging out. He jumps back to the waiting screen in the online multiplayer game he was playing before and realizes that he has not lost anything but some waiting time.

Another example of usage:

Bob logs in and discovers that he has been a victim of three attacks, two of them landed and his ship's condition status has dropped a bit. He now decides to repair his ship to be able to receive more hits without being destroyed. While he plays he discovers that a new attack is incoming. To avoid this attack he randomly teleports to make the incoming missiles lose their lock on him. Bob realizes that he seems to be a target by several other players right now and therefore he upgrades his defensive modules and research to make them better, making it harder to destroy his ship.

### ***3.3 The context/environment in which the system is to be used***

- Since our game will be played during short time intervals, it'll be used anywhere where you have a computer, Internet connection and Mozilla Firefox available. At home will be the most common place for this of course, but other places are for example at school or work during breaks, at a friend's house or perhaps on your laptop when you travel.
- The game will be web based and played through a Mozilla Firefox web-browser, hence no installation needs to be made.

### ***3.4 The scope of the system***

- Inside the scope
  - Log-in system for each user
  - Massive Multi Player
  - Real-time updates of resources for each player
  - Interaction between players (private messaging and through the forum, combat between players)
  - High-score lists
  - Alliances between users (optional for each user)
  - Forums/message boards
  - Game map where all players (online and offline) are displayed
  - Definite goal (when reached, the game ends and a new round starts)
  - Cryptation of user passwords
  - Pictures/Illustrations to make the game feel more lush
- Outside of the scope
  - Animations
  - Chat between online players
  - Any kind of 3D graphics
  - AI
  - Cryptation of information sent between the client and the server
  - Music/Sound Effects

### ***3.5 The main factors that need to be taken in to account when designing and building the system***

- Usability

- For example, there shall be an easy to read and understand, instructions-section on the web-page. Here, users can look up how to do what they want to do. For example a user might want to create a new alliance, but he/she doesn't know how, so she goes to the instructions section and there she gets guided on how to create a new alliance. We will however strive towards an intuitive graphical design of the game
- Intuitive
  - A clear and appealing graphic design of the web page considering our target group. The message board (forum) should be easy to access.
- Lasting appeal
  - The game shall be rewarding by being able to upgrade their ship (for example engines, weapons, shields etc.) and advancing towards the goal of the game (both the point of where the game ends and also to be closer to this goal compared to other players).
- Stability
  - The game should hypothetically not be able to crash and the server shall hopefully be constantly up and running.
- How to get an initial playerbase
  - Because we're making a massive multiplayer online game, the game universe needs to be populated. Some sort of marketing of the game needs to be done
- Keep a flow of new players

It's important that we are aware of that it'll not be possible to keep a high player-base all the time. At some point it will steadily go down until the game is more or less dead. We need to take into account that most web-based games have their peek-period of popularity and can therefore be considered as “trendy”.

### **3.6 Technologies and Risks**

- Technologies (this might change as the project progresses)
  - Javascript and AJAX
  - HTML and CSS
  - SQL
  - PHP
  - Overall risks

In this stage of the project we have come to the conclusion that we will use the following programming languages/techniques/tools: Javascript, AJAX, HTML, CSS, PHP and SQL. Due to the fact that none of the members in the group has ever done web applications on this scale before, there is a medium to high risk that one or more of these languages/techniques/tools won't . These are the decided languages with the following motivations as to why we've chosen them:

#### **3.6.1 Javascript and AJAX**

Javascript and AJAX are both web development techniques. They are mostly used for client side aspects of a page that cannot be written in HTML, for instance, behind the scenes, application code. AJAX stand for Asynchronous Javascript And XML and as the name implies uses Javascript. The difference is that with AJAX, you can create real time updates to the web-page without having to manually update the page. This will be useful for a diverse set of development problems in the game. Imagine for example the basic concept that you would like some variable to be displayed in real-time, then AJAX solves that problem.

**Risks:** Contrary to what you may believe, Javascript is not based on the programming language

Java but is rather designed to have a similar look to Java. What this means remains to be discovered as we've not had any experiences with it or with AJAX, but if problems should arise, the probability that they are threatening to the project is most likely small.

### 3.6.2 HTML and CSS

These languages will be used to solve the problem of structuring and designing the web-page that the game is based around. Basically, everything that the user sees in the web browser will be written in HTML or CSS. CSS is short for Cascading Style Sheets and its practical use is to describe the presentation of a document written in a markup language, that is in our case, HTML. In other words, the game web page will be written in HTML and CSS will be used to make the web page look better.

**Risks:** The main concern here is CSS. The group suffers from lack of knowledge in any extensive practical use of the language. What this mean is that there is an estimated medium to high risk that a big portion of time will have to be spent on learning how to use these languages to make a satisfactory layout and structure of the web-page. This, and the use of HTML is one of the big concerns of the project. However, complete failure in this field is estimated as low. This is because of the abundant amount of information on the Internet concerning these two languages.

### 3.6.3 SQL

The project will have use of a database. All players will have an arbitrary amount of information stored about their accounts individually. We will solve this problem by the use of MySQL which is a database management system.

**Risks:** All members of the group have a firm understanding of large aspects of this problem, from the design of a database, to the implementation of a dialog between database and application. The problem that could arise here is how database management (connecting the database to the web-page) with PHP is implemented. Even though there are large amounts of information about this on the Internet, there are still risks involved. For example, it might take more time than we plan.

### 3.6.4 PHP

This leaves us with the server aspect of the project which will be solved by using PHP. PHP is a general purpose scripting language but has its principal focus on server-side scripting. In short, its practical use will be conversing with the database and delivering web-pages on demand.

**Risks:** Specific members of the group have a good understanding in use of the language. Therefore there's an estimated low risk involved in using this particular language to solve the problem. However, due to the nature of the problem being solved, that is the bulk of the server scripting, a preliminary prediction is that this might develop in to one of the biggest parts of the project.

### 3.6.5 Overall risks:

The group perceives the problem as a fairly simple, however we must take into account all technologies involved. In other words, the problem the group is facing is not in the actual coding itself. We are all fairly good at implementing advanced algorithms in different languages. What we are lacking is a broad know-how of the mentioned technologies/languages above and exactly how they will fill their role in the project. That is, how do you implement a massive multi player web based game? To be certain, no advanced algorithms will be used, but rather a complex network of



simple functions and concepts yet to be discovered.

## 4 Glossary

*Technical definitions taken from Wikipedia.*

AI	Artificial Intelligence means the intelligence that the system demonstrates.
AJAX	Asynchronous JavaScript And XML is a group of inter-related web development techniques used for creating interactive web applications.
Alliances	A formal agreement or treaty between two or more players
Apache	Is a web server.
CMS	Content Management System is a system used to manage the content of a Web site.
CSS	Cascading Style Sheets is a language useful in describing the style of document presentations and used on the client side of the game.
DBMS	DataBase Management System is computer software designed for the purpose of managing databases.
Escape points	Escape points is a measurement on the likelihood in successfully reaching the goal of the game. It's basis is founded on the ship's development.
Game Map	A map that displays game field.
HTML	Hypertext Markup Language is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document.
JavaScript	Is a scripting language most often used for client-side web development.
MediaWiki	Is a web-based wiki software application used as the project group's CMS.
Modules	Attachable components to the ship.
Mozilla Firefox	Is a web browser with 16.01% of the recorded market share in Web browsers as of November 2007.
MySQL	Is multithreaded, multi-user SQL DBMS.
PHP	Is a reflective computer programming language designed for producing dynamic web pages.
Resources	Resources are the currency which the players can use.
Skills Star	These are points that allows the players to specialize their research within different areas.

AI	Artificial Intelligence means the intelligence that the system demonstrates.
SQL	Is a computer language designed for the retrieval and management of data in relational database management systems, database schema creation and modification, and database object access control management.
SVN	Subversion, often abbreviated SVN, is a version control system that allows users to keep track of changes made to any type of electronic data, typically source code, web pages or design documents.
Wiki	Is software that allows users to create, edit, and link web pages easily and is used as a CMS in the project.
World Wide Web Consortium	Is the main international standards organization for the World Wide Web. Often abbreviated to W3C.
XHTML	Extensible HyperText Markup Language, or XHTML, is a markup language that has the same depth of expression as HTML, but also conforms to XML syntax.
XML	Extensible Markup Language (XML) is a general-purpose markup language.

# 5 User requirements definition

## Table of contents

- 5 User requirements definition..... 12
  - 5.1 Functional requirement..... 12
    - 5.1.1 Web page..... 12
    - 5.1.2 Game map..... 13
    - 5.1.3 Resources..... 13
    - 5.1.4 The ship..... 13
    - 5.1.5 Modules..... 14
    - 5.1.6 High-Score lists..... 15
    - 5.1.7 Alliances..... 15
    - 5.1.8 Movement..... 15
    - 5.1.9 Communication..... 15
    - 5.1.10 Research..... 15
  - 5.2 Non-functional requirements..... 16
    - 5.2.1 Product requirements..... 16
      - 5.2.1.1 Usability requirements..... 16
      - 5.2.1.2 Performance requirements..... 16
      - 5.2.1.3 Portability requirements..... 17
    - 5.2.2 Organisational requirement..... 17
      - 5.2.2.1 Delivery requirements..... 17
      - 5.2.2.2 Implementation requirements..... 17
    - 5.2.3 External requirements..... 18
      - 5.2.3.1 Interoperability requirements..... 18
        - 5.2.3.1.1 Ethical requirements..... 18
        - 5.2.3.1.2 Privacy requirements..... 18
      - 5.2.3.2 Safety requirements..... 18

### 5.1 Functional requirement

#### 5.1.1 Web page

<b>5.1.1.1</b>	Visitors shall be able to create an account.
<b>5.1.1.2</b>	Log in with an existing account.
<b>5.1.1.3</b>	Read instructions about the game.
<b>5.1.1.4</b>	Visit the forum.

## 5.1.2 Game map

<b>5.1.2.1</b>	<p>The game map will be a coordinate system where the players can move freely. There will be another layer placed upon this coordinate system which will be a grid field.</p>
<b>5.1.2.2</b>	<p>There is a wormhole located at the origin of the game map. The goal of the game is to reach it. When the player succeeds in this, he/she wins the current game round and then a new round starts up where all players will be assigned a default ship.</p>
<b>5.1.2.3</b>	<p>All players will start at an equidistance from the origin of the game map. This will result in all players starting in a circle.</p> <p>New players will start near each other.</p> <p>When a new round starts up, if the amount of players are not enough to be divided around the whole circle, i.e. they are too far away from each other, they will be spread out over a part of the circle.</p> <p>A grid square on the map can only be occupied by one player at a time.</p>
<b>5.1.2.4</b>	<p>Each player will be able to view the entire map and all other players by moving the map view.</p>

## 5.1.3 Resources

<b>5.1.3.1</b>	<p>There are two types of resources. The main one is obtained by building power plant modules and the other one can be gathered from resource squares on the game map at the cost of the the main resource.</p> <p>Players can use their resources to improve their ship.</p>
<b>5.1.3.2</b>	<p>The resource squares will be identical to the other squares on the main map apart from the fact that the player can gather the additional resource and the appearance of the squares.</p>

## 5.1.4 The ship

<b>5.1.4.1</b>	<p>Each player start with their own default ship and cannot under any circumstances gain another one.</p> <p>The default ship only contain one power plant module (more on this under “modules”)</p>
<b>5.1.4.2</b>	<p>When the condition status of the ship reaches zero percent, the ship will be destroyed and the player has been defeated. The player will then be assigned a new default ship.</p>

<b>5.1.4.3</b>	If damaged, the ship's condition status can be improved in exchange for a certain value of resources. The pace in which the ship rebuilds condition status will depend on the repair module/modules that the player has and the amount of resources spent.
<b>5.1.4.4</b>	Movements of the ship cost resources and it takes time to travel from one place to another.

## 5.1.5 Modules

Modules can be attached to the ship for an exchange of resources. All modules except for the power plant module needs to be researched before the player can build them. Each module can be upgraded to another level. These are the following modules which can be added:

<b>5.1.5.1</b>	<b>Offensive weapons are used to attack players.</b>
<b>5.1.5.1.1</b>	Missile batteries module: Missiles can reach any player on the entire game map. However, it takes time for the missile to travel through space. Missile battery modules may launch a fixed number of missiles at a time. They can store a certain number of missiles . For each level this module is upgraded, the storage of missiles is expanded and the amount of missiles fired at a time is increased.
<b>5.1.5.1.2</b>	Cannons module: Cannon shells can only reach players close to the ship. There is a delay time before impact, i.e. it takes time for the projectile to travel through space. Cannon modules may fire a fixed number of shells at a time. They can store a certain number of shells. For each level this module is upgraded, the storage of shells is expanded and the player can fire more shells at a time.
<b>5.1.5.2</b>	<b>Defensive weapons are used to avoid attacks.</b>
<b>5.1.5.2.1</b>	The teleportation module moves the ship to a random grid square within a short distance. This can be used to avoid incoming missiles and canon attacks from close players. It costs a lot more of the main resource to use teleportation than to actually travel the equal distance. For each level this module is upgraded, the costs when using this module decreases. Teleportation can also be used on other players ships but teleportation has a certain range in which it can be used.
<b>5.1.5.2.2</b>	The missile decoy module are used to avoid incoming missiles by producing missile decoys to intercept them. For each level this module is upgraded the overall capacity of decoys deployed at a time is increased and the production capacity of decoys is increased.
<b>5.1.5.3</b>	Storage modules: Is used to store resources. The amount of resources that can be stored depends on the level of this module.
<b>5.1.5.4</b>	The engine module. The level of the engines determines how fast the ship can travel to a specified coordinate.

<b>5.1.5.5</b>	Power plant module that produces the main resource. All players starts with one power plant that constantly generates the main resource. When upgrading the power plant module, the player receives more resources over time.
<b>5.1.5.6</b>	The repair module. The repair module repairs the ship for an exchange for recourses. By upgrading this module, the ship repairs faster.

### 5.1.6 High-Score lists

There will be two types of lists:

<b>5.1.6.1</b>	A list that shows players sorted after distances from the center of the map (the wormhole) in falling order.
<b>5.1.6.2</b>	A high-score list of players' Escape points (More on this under "Escape Points").
<b>5.1.6.3</b>	Escape points will be gained when the player adds modules, upgrades modules or researches (more on this under "Research"). Each time a player gains a certain amount of escape points, he/she will be awarded stars that can be appointed on different skills. Each star the player appoints on a skill decreases the time it takes to research in the corresponding category.

### 5.1.7 Alliances

<b>5.1.7.1</b>	Players will be able to form alliances with each other, but there can only be one winner.
<b>5.1.7.2</b>	The only benefit from being in an alliance is that other players can see that you are in an alliance, and also what other players are also in it. It is not possible to trade resources.

### 5.1.8 Movement

<b>5.1.8.1</b>	The ships will be able to move around in any direction on the game map. The grid will be used for identifying if two ships are in the same area. There will therefore be coordinates within a grid square.
----------------	--

### 5.1.9 Communication

<b>5.1.9.1</b>	All players will be able to send short private text messages to each other.
<b>5.1.9.2</b>	There will also be a forum on the main web-page where players can discuss various topics.

### 5.1.10 Research

This list depicts which fields can be researched, and what benefit the player gains from them:

<b>5.1.10.1</b>	Missiles: decreases the time it takes for missiles to travel to another player. Increases the accuracy of the missiles.
-----------------	---

<b>5.1.10.2</b>	Cannons: increases the range of the weapon and increases the damage done.
<b>5.1.10.3</b>	Missile decoys: increases the interception rate of the decoys.
<b>5.1.10.4</b>	Teleportation: increases the range in which you can use it on other players.
<b>5.1.10.5</b>	Engines: decreases the costs of resources when traveling through space.
<b>5.1.10.6</b>	Repair: decreases the cost of repairing the ship.

## **5.2 Non-functional requirements**

### **5.2.1 Product requirements**

#### **5.2.1.1 Usability requirements**

<b>5.2.1.1.1</b>	It shall be easy to create and login on the game's website so the player can get quickly into the game
<b>5.2.1.1.2</b>	All text shall be written in English since most potential players understand the language.
<b>5.2.1.1.3</b>	Extensive documentation about how to play the game shall be included in the first version of the game.
<b>5.2.1.1.4</b>	The graphical user interface shall be based on XHTML, CSS and JavaScript. This type of GUI makes it possible for the player to access the game from different computers as long as the computer has the web browser Mozilla Firefox.

#### **5.2.1.2 Performance requirements**

<b>5.2.1.2.1</b>	The game will be designed for Mozilla Firefox and, therefore, it is likely that the recommended system requirements for Mozilla Firefox are enough to play the game. Different web browsers use different standards of the programming languages used. In other words, it would take too much time implement the game for different web browsers and therefore we've chosen to make the game for Mozilla Firefox.
<b>5.2.1.2.2</b>	The game is a multiplayer game and, therefore, the server has to be connected to the internet. Hence, the players also need to have internet access to connect with their web browser to the server.



### 5.2.1.3 Portability requirements

5.2.1.3.1	The game shall be possible to play with the specified web browser.
-----------	--

## 5.2.2 Organisational requirement

### 5.2.2.1 Delivery requirements

5.2.2.1.1	The developers must have time to complete the design document before the deadline for the final version.
5.2.2.1.2	The developers must have time to complete the game before the deadline for the final version.
5.2.2.1.3	The first beta version of the game shall be provided to Rand Waltzman on specified date.

### 5.2.2.2 Implementation requirements

5.2.2.2.1	There is no time to implement the game on different servers. Besides, the game will not be sold to customer; therefore, it's no need to provide support for more than one server. Apache is open source and, therefore, free to use for the developers.
5.2.2.2.2	The game will use the specified database management system (DBMS). MySQL is one of the most popular database management systems for web based software. It is also open source and, therefore, free to use for the developers.
5.2.2.2.3	The specified programming language shall be used to implement the game on the web server. The developers have moderate experience of the programming language. It is used for enterprise web solutions. It is also open source and, therefore, free for the developers to use. PHP is also supported on web hosting providers.
5.2.2.2.4	On the client side of the game: Extensible HyperText Markup Language (XHTML), Cascading Style Sheets (CSS) and JavaScript will be used to implement the graphical user interface (GUI) of the game. All three languages/tools work with the specified web browser and the developers have moderate knowledge about them.
5.2.2.2.5	During the development of the game a version control system shall be used to keep track of the development. SVN is an open source version control system that will be used.
5.2.2.2.6	During the development of the game all documentation shall be saved on the content management system (CMS) specified. MediaWiki is easy to use, even for beginners, and it works in different web browsers. It is easy to organize and keep track of different pages in the system. MediaWiki is open source and, therefore, free to use.

## 5.2.3 External requirements

### 5.2.3.1 Interoperability requirements

#### 5.2.3.1.1 Ethical requirements

<b>5.2.3.1.1.1</b>	No offensive language and pictures in the description of the game, the game's objectives. Younger players should be able to play the game. It is outside the scope of the project to develop tools to scan for offensive language and pictures in the communication between players.
--------------------	--

#### 5.2.3.1.2 Privacy requirements

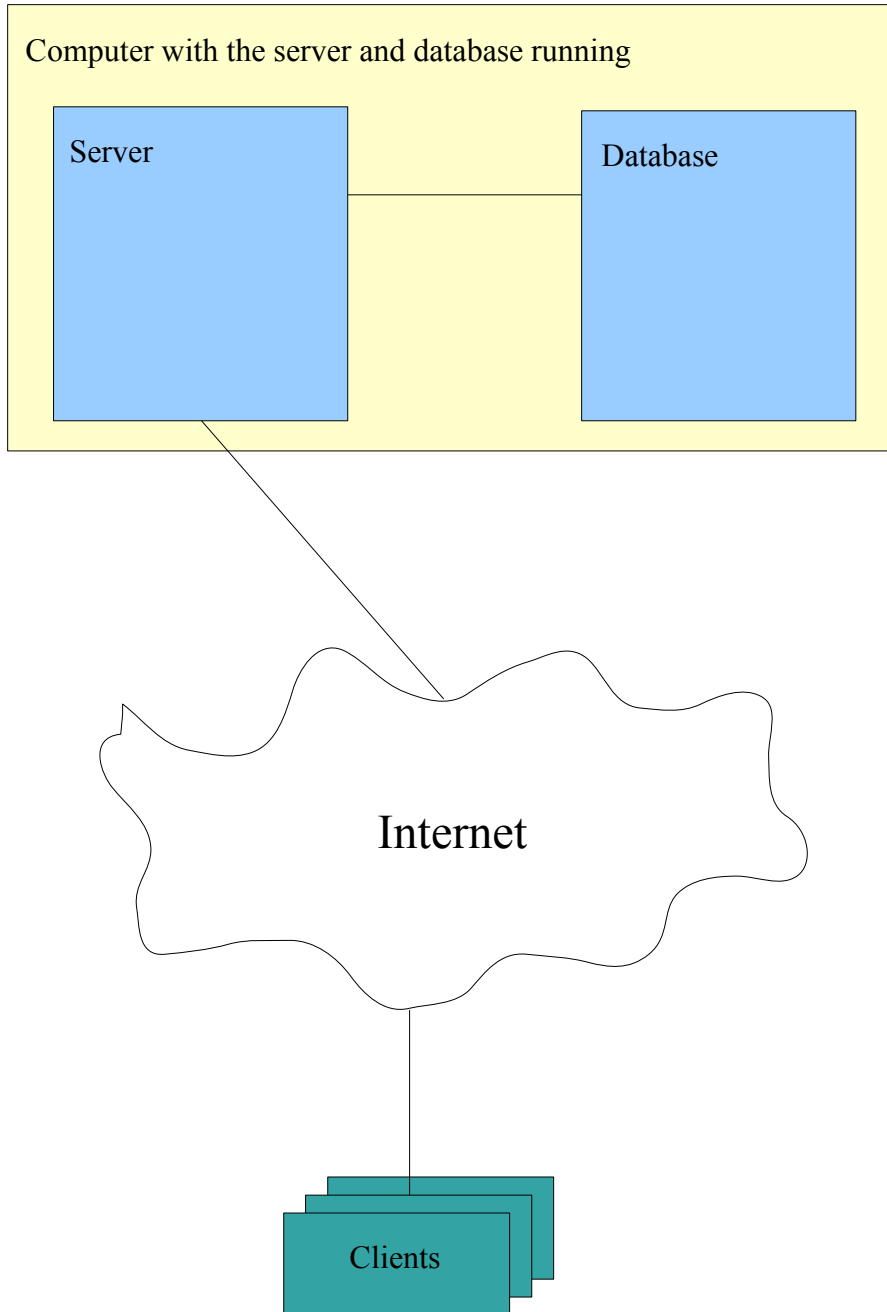
<b>5.2.3.1.2.1</b>	The game is likely to be hosted on a Swedish web server and, therefore, the game has to follow Swedish regulations of using cookies.
--------------------	--

#### 5.2.3.2 Safety requirements

<b>5.2.3.2.1</b>	No personal information about the player will be saved except username, password and e-mail. The web host providers will save the information.
<b>5.2.3.2.2</b>	The game will not provide a secure connection, i.e. the data transferred between the server and the client will not be encrypted. Not even when the players logs in to the game. The developers don't have time to implement the use of secure connection between the server and the client of the game.

## 6 System architecture

High level overview of the anticipated system architecture:



The clients will be mozilla firefox web-browsers used by the people that play our game, they communicate with the server through internet. The server then stores and collects information from the database which will run on the same machine as the server.

# 7 System requirements specification

## Table of contents

7 System requirements specification.....	20
7.1 Functional requirements.....	21
7.1.1 Web page.....	21
7.1.1.1 Create an account.....	21
7.1.1.2 Log in.....	22
7.1.1.3 Game instructions.....	22
7.1.1.4 Forum.....	22
7.1.2 Game map.....	22
7.1.2.1 Game map.....	22
7.1.2.2 The wormhole.....	22
7.1.2.3 Startup placement of players.....	23
7.1.2.4 Player view.....	23
7.1.3 Resources.....	23
7.1.3.1 Main resource.....	23
7.1.3.2 Secondary resource.....	23
7.1.3.3 Resource squares.....	23
7.1.4 The ship.....	24
7.1.4.1 Default ship.....	24
7.1.4.2 Destroying ship.....	24
7.1.4.3 Damage ship.....	24
7.1.4.4 Repair ship.....	24
7.1.4.5 Movements of ship.....	25
7.1.5 Modules.....	25
7.1.5.1 Offensive Weapons.....	25
7.1.5.1.1 Missile batteries module.....	25
7.1.5.1.2 Cannons module.....	26
7.1.5.2 Defensive Weapons.....	27
7.1.5.2.1 Teleportation module.....	27
7.1.5.2.2 Missile decoys module.....	27
7.1.5.3 Storage module.....	28
7.1.5.4 Engine module.....	29
7.1.5.5 Power Plant.....	29
7.1.5.6 Repair module.....	29
7.1.6 High-score list.....	29
7.1.6.1 Escape points.....	29
7.1.6.2 Skills star awards.....	29
7.1.7 Alliances.....	30
7.1.7.1 Players in an alliance.....	30
7.1.8 Communication.....	30
7.1.8.1 Text messages.....	30
7.1.8.2 Forum.....	30
7.1.9 Research.....	30
7.1.9.1 Researching missiles.....	30

7.1.9.2 Researching cannons.....	31
7.1.9.3 Researching missile decoys.....	31
7.1.9.4 Researching teleportation.....	31
7.1.9.5 Researching engines.....	31
7.1.9.6 Researching repair.....	32
7.2 Non-functional requirements.....	32
7.2.1 Product requirements.....	32
7.2.1.1 Usability requirements.....	32
7.2.1.1.1 Login.....	32
7.2.1.1.2 Language.....	32
7.2.1.1.3 Documentation in the game.....	32
7.2.1.1.4 Client user interface.....	33
7.2.1.2 Performance requirements.....	33
7.2.1.2.1 Client side performance.....	33
7.2.1.2.2 Network.....	33
7.2.1.3 Portability requirements.....	33
7.2.1.3.1 Web browsers.....	33
7.2.2 Organisational requirements.....	34
7.2.2.1 Delivery requirements.....	34
7.2.2.1.1 Delivery of the requirement document.....	34
7.2.2.1.2 Delivery of design document.....	34
7.2.2.1.3 Delivery of first version.....	34
7.2.2.2 Implementation requirements.....	34
7.2.2.2.1 Web server software.....	34
7.2.2.2.2 Database management system.....	35
7.2.2.2.3 Server side implementation language.....	35
7.2.2.2.4 Client side implementation languages.....	35
7.2.2.2.5 Version control system.....	35
7.2.2.2.6 Documentation system.....	36
7.2.3 External requirements.....	36
7.2.3.1 Interoperability requirements.....	36
7.2.3.1.1 Ethical requirements.....	36
7.2.3.1.2 Privacy requirements.....	36
7.2.3.2 Safety requirements.....	36
7.2.3.2.1 Information saved about the user.....	36
7.2.3.2.2 Secure connection.....	37

## **7.1 Functional requirements**

### **7.1.1 Web page**

#### **7.1.1.1 Create an account**

**Description:** The player shall have an account

**Rationale:** The users need to identify themselves so they only can control their own ship.

**Goal:** The player shall create an account.

**Input:** Desired user name and password.

**Error conditions:**

The user name can be taken by someone else.

**Behaviors:**

If the user name is taken by someone else the user will be prompted to choose another user name.

**Output:** A new account.

**Distributed across the following system modules:** Server and Database

### **7.1.1.2 Log in**

**Description:** The players need to log in to identify themselves.

**Rationale:** The game would be chaotic if everyone could control anyone's ship.

**Goal:** The player shall log on to his account.

**Input:** User name and password.

**Error conditions:**

The user name and password does not match.

**Behaviors:**

If user name and password does not match the user will be prompted to enter the information again.

**Output:** Access to the players ship.

**Distributed across the following system modules:** Server and Database

### **7.1.1.3 Game instructions**

**Description:** People can read about the game.

**Rationale:** It will be easier to get people playing the game if it can be quickly learned.

**Distributed across the following system modules:** Server and Database

### **7.1.1.4 Forum**

**Description:** People can discuss the game.

**Rationale:** The players will most likely have questions, discuss strategies and want to brag to each other.

**Distributed across the following system modules:** Server

## **7.1.2 Game map**

### **7.1.2.1 Game map**

**Description:** The game map will be a coordinate system where the players can move freely. There will be another layer placed upon this coordinate system which will be a grid field.

**Rationale:** The grid field placed upon the coordinate system will be used to check if there are more than one player in the same area. The players will be able to freely move around the map in any direction, as long as there is no ship nearby. This brings forth the ability to block the way for other players.

**Distributed across the following system modules:** Client, Server and Database

### **7.1.2.2 The wormhole**

**Description:** The square in the middle will display the wormhole

**Rationale:** For the players to know where the goal is located.

**Distributed across the following system modules:** Server and Database

### **7.1.2.3 Startup placement of players**

**Description:** All players will start at an equidistance from the origin of the game map.

**Rationale:** Because of the goal of the game, where the goal is to reach the center of the map, it is important to place all players with equal distance from the center, so all the players have the same preconditions in the game. If there are few players they will be grouped together.

**Input:** Number of players

**Behaviors:** Many players will result in equal distance between them all, fewer players will be grouped together to avoid long distances.

**Output:** Placement of players

**Distributed across the following system modules:** Server and Database

### **7.1.2.4 Player view**

**Description:** All players should be able to see every other player as well as all objects e.g. resources on the map.

**Rationale:** For the attack system to work, where each player can attack any other player on the field, all players need to be able to see each other. For the players to make active decisions of whether they are to gather resources or not, the map will show the resources present on the map. When the resources in the square is empty, the square will become a regular square again.

**Goal:** Make the players aware of their surroundings.

**Distributed across the following system modules:** Server and Database

## **7.1.3 Resources**

### **7.1.3.1 Main resource**

**Description:** The main resource is meant to be an energy type resource.

**Rationale:** The energy type resource will be used in all scenarios where the use of energy instead of minerals is more logical.

**Behaviors:**

Obtained by building power plant modules.

Can be used to improve the ship, move the ship, and gather the secondary resource.

**Distributed across the following system modules:** Server and Database

### **7.1.3.2 Secondary resource**

**Description:** The secondary resource is meant to be a mineral based resource.

**Rationale:** The minerals will be used in all scenarios where the use of minerals instead of energy is more logical.

**Behaviors:**

Obtained by standing on a resource square at the cost of the main resource.

Can be used to improve the ship.

**Distributed across the following system modules:** Server and Database

### **7.1.3.3 Resource squares**

**Description:** The secondary resource can be gathered in special locations.

**Rationale:** It'll be harder to come by this type of resource and players will fight over the resource

squares.

**Goal:** To make the game more interesting.

**Behaviors:**

The resource squares will be identical to the other squares on the map apart from the fact that the player can gather the secondary resource and the appearance of the square.

**Distributed across the following system modules:** Server and Database

## 7.1.4 The ship

### 7.1.4.1 Default ship

**Description:** The default ship is assigned to a new player or a player who has lost his or hers ship. The player shall only have one ship.

**Rationale:** All players shall start with the same ship.

**Goal:** The player shall obtain a new default ship.

**Output:** Default ship contains one power plant, the ability to gather the secondary resource and the ability to move.

**Distributed across the following system modules:** Client, Server and Database

### 7.1.4.2 Destroying ship

**Description:** The ship can be destroyed.

**Rationale:** It is more interesting if players are able to completely defeat other players.

**Goal:** The ship's conditions status reach zero.

**Output:** The player will be assigned a new default ship.

**Distributed across the following system modules:** Client, Server and Database

### 7.1.4.3 Damage ship

**Description:** The ship can be damaged.

**Rationale:** A ship should not be destroyed by one hit, therefore ships have the ability to take damage and survive.

**Goal:** The ship's condition status shall be decreased.

**Input:** Damage done by weapon.

**Error conditions:** No damage done.

**Behaviors:**

The ship's condition status will be decreased.

**Output:** The ship's condition status is decreased.

**Distributed across the following system modules:** Client, Server and Database

### 7.1.4.4 Repair ship

**Description:** The ship shall be able to repair itself.

**Rationale:** When the ship is damaged in battle it must be able to repair itself.

**Goal:** The ship's condition status shall increase.

**Input:** Main resource

**Error conditions:** Not enough resources.

**Behaviors:**

The amount of repair modules (see more in the section Modules) affects the pace the ship regeneration rate.



If there is no resources the ship's condition status will not increase.

**Output:** The ship's condition status shall be increased.

**Distributed across the following system modules:** Server and Database

#### **7.1.4.5 Movements of ship**

**Description:** Ship shall be able to move.

**Rationale:** To reach the goal of the game the ship has to be able to move.

**Goal:** To move from one place to another.

**Input:** The coordinates where the ship is and where the ship shall travel, ship's engines, research on engines.

**Error conditions:**

Not enough resources.

Another ship is within the grid square that includes the desired destination the ship will stop as close as it can.

**Behaviors:**

It takes time to move the ship.

The level of engine modules determines how fast the ship will travel to a specified coordinate.

Engine research decreases the cost of the main resource to travel.

**Output:** The ship has reached the desired set of coordinates

**Distributed across the following system modules:** Client, Server and Database

### **7.1.5 Modules**

**Description:** Modules can be attached to the ship for an exchange of varying resources. Several modules of the same type can be attached to the ship. All modules except for the power plant module needs to be researched in before the player can build them. Each module can be upgraded to another level. These are the following modules which can be added:

#### **7.1.5.1 Offensive Weapons**

**Description:** Offensive Weapons are used to attack other players.

##### **7.1.5.1.1 Missile batteries module**

**Description:** Missiles will be effective because of the fact that they can reach any player on the entire game map. In order to balance this out it will take time for missiles to travel through space and the accuracy will depend on the researches done on this module. The player will also have to build missiles before firing them away and he/she can only fire a fix amount of missiles at a time.

###### **7.1.5.1.1.1 Storage of missiles**

**Description:** The amount of missiles that can be stored depends on the level of the missile batteries module.

**Rationale:** There will be a fix amount of missiles that each missile battery module can store. This is crucial for the balance of the game since it limits the amount of missiles that each player can build, in order to build a certain amount of missiles the player needs to be able to store them. The fact that the maximum amount of storage of missiles varies depending on the level of the module gives the player more satisfaction when upgrading the module.

**Distributed across the following system modules:** Server and Database

#### 7.1.5.1.1.2 Building missiles

**Description:** Description: The amount of missiles that the player can produce at a time depends on the player's resources and how many missiles the player can store

**Rationale:** It shall cost to build missiles, otherwise missiles could just be built automatically. This makes missiles more exquisite and will increase the overall balance of the game due to the fact that it will reduce the amount of missiles that will be fired.

**Goal:** Build missiles

**Input:** An amount of missiles and resources

**Error Conditions:** Missiles fails to be built

**Behaviours:**

Missiles are being built

Missiles fails to be built due to insufficient amount of storage

Missiles fails to be built due to insufficient amount of resources

**Output:** Missiles produced and stored

**Distributed across the following system modules:** Client, Server and Database

#### 7.1.5.1.1.3 Launch missiles

**Description:** A player can fire missiles but it will take time before he/she can fire another round of missiles. That time will not change by upgrading the module. What will change though is the amount of missiles that can be launched at a time. It will take time for the missile to travel through space. This will decrease as this module is researched.

**Rationale:** A player should be able to launch missiles

**Distributed across the following system modules:** Client, Server and Database

#### 7.1.5.1.2 Cannons module

**Description:** The cannons module will be effective in close ranged combats.

##### 7.1.5.1.2.1 Storage of shells

**Description:** In order to fire the cannons, shells need to be built. The amount of shells that can be stored depends on the level of this module.

**Rationale:** The player cannot store an infinite amount of shells. Se motivation for storing missiles

**Distributed across the following system modules:** Client, Server and Database

##### 7.1.5.1.2.2 Building shells

**Description:** The amount of shells that the player can build will depend on how much resources he/she has and the level of this module. Similar to the missiles batteries, it will cost resources to build ammunition.

**Rationale:** It should not be free to damage other players.

**Goal:** Build shells

**Input:** An amount of shells and resources

**Error Conditions:** Shells fails to be built

**Behaviours:**

Shells are being built

Shells fails to be built due to insufficient amount of storage

Shells fails to be built due to insufficient amount of resources

**Output:** Shells produced, stored and ready for disposal

**Distributed across the following system modules:** Client, Server and Database

### 7.1.5.1.2.3 Firing cannons

**Description:** A player can fire the cannons but there will be a delay time before impact since the shells need to travel through space. For each level the cannon module is upgraded, the more shells a player can fire at a time.

**Rationale:** The player shall be able to fire the shells.

**Goal:** Fire cannons

**Input:** A targeted ship

**Error Conditions:** Cannons fails to be fired

**Behaviours:**

Cannons are being fired

The targeted ship is outside of the cannons range resulting in the cannons not being fired.

Player has an insufficient amount of ammunition

**Output:** Cannons are fired and the time before impact starts ticking

**Distributed across the following system modules:** Client, Server and Database

### 7.1.5.2 Defensive Weapons

**Description:** Defensive weapons are used to defend from attacks from other ships

#### 7.1.5.2.1 Teleportation module

**Description:** The teleportation module moves the ship to a random grid square within a short distance. This can be used to avoid incoming missiles and canon attacks from close players. It costs lot more of the main resource to use teleportation than to actually travel the equal distance. For each level this module is upgraded, the costs when using this module decreases. Teleportation can also be used on other players but teleportation has a certain range in which it can be used.

**Rationale:** The main function of the teleportation module is to escape canon or missile attacks.

**Goal:** Teleport oneself

**Input:** Resources

**Error Conditions:** Failure of teleportation

**Behaviours:**

Ship is being teleported to a random set of coordinates

The player has an insufficient amount of resources for teleportation

**Output:** Movement through space

**Distributed across the following system modules:** Client, Server and Database

#### 7.1.5.2.2 Missile decoys module

**Description:** The missile decoy module is used to avoid incoming missiles by intercepting them. The player can build missile decoys in this module and the production capacity of missile decoys depends on the level of this module. The overall capacity of decoys depends on the level of this module.

**Rationale:** This is a necessary module due to the fact that missiles can reach any player on the entire map and will therefore most likely be the most common weapon that players choose. Teleportation also works against missiles but missile decoys are much cheaper to produce and are a passive defense in a sense that they automatically intercept incoming missiles at a given

##### 7.1.5.2.2.1 Capacity of decoys

**Description:** The capacity of decoys depends on the level of the missile decoy module

**Rationale:** There needs to be a limitation to the amount of missile decoys each player can have.

Otherwise it may be too easy to defend against missiles.

**Distributed across the following system modules:** Server and Database

#### **7.1.5.2.2 Building decoys**

**Description:** The amount of decoys that players can build will depend on their amount of resources and the capacity of the module.

**Rationale:** It shall cost to build missile decoys and each player should not be able to produce an infinite amount of decoys.

**Goal:** Build decoys

**Input:** An amount of decoys and resources

**Error Conditions:** Decoys fails to be built

**Behaviours:**

Decoys are being built

Decoys fails to be built due to insufficient amount of capacity

Decoys fails to be built due to insufficient amount of resources

**Output:** Decoys produced

**Distributed across the following system modules:** Client, Server and Database

#### **7.1.5.2.3 Nullifying missile attack**

**Description:** Missile decoys nullifies incoming missiles automatically at a given success rate.

**Rationale:** Missile decoys can intercept missiles

**Goal:** Nullify missile attack

**Input:** Percentage of interception

**Error Conditions:** Decoy fails to intercept missile

**Behaviors:**

Decoy intercepts missile

Decoy fails to intercept missile due to the percentage of interception

**Output:** Attack nullified

**Distributed across the following system modules:** Client, Server and Database

#### **7.1.5.3 Storage module**

**Description:** There are two kinds of storage modules. One storage module is used for storing the additional resource that can be gathered from the resource squares and the other is used to store the main resource that is produces in the power plants. The amount of resources that can be stored depends on the level of the module.

**Rationale:** The player shall not be able to gather an infinite amount of the additional resource. This will be limited by including a storage module.

**Goal:** Store resources

**Input:** Resources

**Error Conditions:** Resources fails to be stored

**Behaviors:**

Resources are being stored

Resources fails to be stored due to insufficient amount of space on the storage module

**Output:** Resources stored

**Distributed across the following system modules:** Server and Database

#### **7.1.5.4 Engine module**

**Description:** The level of the engines determines how fast the ship can travel to a specified coordinate

**Rationale:** Players shall be able to decrease the time it takes to travel through space.

**Distributed across the following system modules:** Server and Database

#### **7.1.5.5 Power Plant**

**Description:** The power plant module increases the production of the main resource. The main resource is produced automatically. All players starts with one power plant. When upgrading the power plant module, the player receives more resources over time.

**Rationale:** The main resource shall be constantly ticking in for the player.

**Distributed across the following system modules:** Server and Database

#### **7.1.5.6 Repair module**

**Description:** The repair module repairs the ship for an exchange of resources. Reparation of the ship is made over time and automatically. By upgrading this module, the ship repairs faster.

**Rationale:** The reparation module is crucial for players to recover from attacks from their opponents.

**Distributed across the following system modules:** Server and Database

### **7.1.6 High-score list**

**Description:** There will be two types of high-score lists. One that shows the players sorted after distances from the center in falling order and another one that bases upon escape points. Escape points are based on how much resources the player has spent on modules and research.

**Raionale:** High-score lists acknowledges peoples' efforts and encourages people to compete with each other.

**Distributed across the following system modules:** Server and Database

#### **7.1.6.1 Escape points**

**Description:** Escape points will be gained when adds modules, upgrades modules or researches, and the amount of points awarded will correspond to the amount of resources invested in each investment. All players will be able to see all other players escape points.

**Rationale:** Players will want to check their status in the game world. This will enable them too see who is the strongest when it comes to modules and research.

**Goal:** Display players Escape points.

**Input:** Amount of resources spent on modules and research.

**Error condition:** None

**Behaviors:** Escape point calculation.

**Output:** Escape point list.

**Distributed across the following system modules:** Server and Database

#### **7.1.6.2 Skills star awards.**

**Description:** Each time a player gains a certain amount of escape points, he/she will be awarded stars that can be appointed on different skills. Each star the player appoints on a skill decreases the time it takes to research in the corresponding category. One amount of stars will be given to each player at the start of the game round. Each skill in which a star can be appointed can hold a

maximum of stars. When that maximum is reached the player must choose another skill to invest stars in.

**Rationale:** Appointing stars in different research fields will make players even more specialized.

**Goal:** Specialize in a research field.

**Input:** Escape points.

**Error conditions:** None

**Behaviors:** Stars can be appointed in missiles, cannons, decoys, teleportation, engines and repair.

**Output:** Appointed stars.

**Distributed across the following system modules:** Client, Server and Database

## 7.1.7 Alliances

### 7.1.7.1 Players in an alliance

**Description:** Players can form alliances with each other. There may be an unlimited number of players in an alliance. If a player of an alliance wins, the alliance will not receive any credit. A player can only be in one alliance at a time.

**Rationale:** The teamwork part of the game will bring in the element of strategic gaming. But because only one player can win the whole game, the internal fight will bring forth even another aspect of the game play, namely the fight internally in the alliance.

**Distributed across the following system modules:** Client, Server and Database

## 7.1.8 Communication

### 7.1.8.1 Text messages.

**Description:** The player should be able to send short text messages to other players.

**Rationale:** To give the players an ability to communicate.

**Distributed across the following system modules:** Client, Server and Database

### 7.1.8.2 Forum

**Description:** The players will have access to a forum where they can discuss the game.

**Rationale:** A simple forum where the players can post and discuss about the game is another way of keeping the players coming back to the website and keeping the interest for the game when not playing.

**Distributed across the following system modules:** Client, Server and Database

## 7.1.9 Research

**Description:** All players will be able to research in different fields.

**Rationale:** By researching in different fields the player is allowed to specialize the field that the Player feels will be most beneficial.

**Distributed across the following system modules:** Client, Server and Database

### 7.1.9.1 Researching missiles

**Description:** Decreases the time it takes for missiles to travel to another player. Increases the accuracy of the missiles.

**Rationale:** Player should be able to specialize in missile weapon technology if they feel it will further their cause.

**Goal:** Improve missiles.

**Input:** Resources.

**Error condition:** Player fails to research missiles.

**Behaviors:** Researching missiles, Player has insufficient amount of resources.

**Output:** Missile research.

**Distributed across the following system modules:** Client, Server and Database

#### ***7.1.9.2 Researching cannons***

**Description:** Increases the range of the weapon and increases the damage done by the weapon.

**Rationale:** Player should be able to specialize in cannon weapon technology if they feel it will further their cause.

**Goal:** Improve Cannons.

**Input:** Resources.

**Error condition:** Player fails to research cannons.

**Behaviors:** Research cannons, Player has insufficient amount of resources.

**Output:** Cannon research.

**Distributed across the following system modules:** Client, Server and Database

#### ***7.1.9.3 Researching missile decoys.***

**Description:** Increases the interception rate of the decoys.

**Rationale:** Player should be able to specialize in missile defensive weapon technology if they feel it will further their cause.

**Goal:** Improve missile decoys.

**Input:** Resources.

**Error condition:** Player fails to research missile decoys.

**Behaviors:** Researching missile decoys, Player has insufficient amount of resources.

**Output:** Missile decoy research.

**Distributed across the following system modules:** Client, Server and Database

#### ***7.1.9.4 Researching teleportation***

**Description:** Increases the range in which you can use teleportation on other players.

**Rationale:** Player should be able to specialize teleportation technology if they feel it will further their cause.

**Goal:** Improve teleportation.

**Input:** Resources.

**Error condition:** Player fails to research teleportation.

**Behaviors:** Researching teleportation, Player has insufficient amount of resources.

**Output:** Teleportation research.

**Distributed across the following system modules:** Client, Server and Database

#### ***7.1.9.5 Researching engines***

**Description:** Decreases the cost of resources when traveling through space.

**Rationale:** Player should be able to specialize in engines technology if they feel it will further their cause.

**Goal:** Improve engines.

**Input:** Resources.

**Error condition:** Player fails to research engines.

**Behaviors:** Researching engines, Player has insufficient amount of resources.

**Output:** Engine research.

**Distributed across the following system modules:** Client, Server and Database

#### **7.1.9.6 Researching repair**

**Description:** Decreases the cost of ship repair.

**Rationale:** Player should be able to specialize in repair technology if they feel it will further their cause.

**Goal:** Improve repair.

**Input:** Resources.

**Error condition:** Player fails to research repair improvement.

**Behaviors:** Researching repair improvement, Player has insufficient amount of resources.

**Output:** Repair research.

**Distributed across the following system modules:** Client, Server and Database

## **7.2 Non-functional requirements**

### **7.2.1 Product requirements**

#### **7.2.1.1 Usability requirements**

##### **7.2.1.1.1 Login**

**Description:** It shall be easy to create and login on the game's website.

**Rationale:** It is important that the player can start to play the game fast.

**Unit:** Minutes

**Measure:** The period of time it takes from creating an account to the player has logged in on the web page.

**Goal:** 3 minutes

**Side effects:** The game's objective and how to play must be briefly described after the players have created their accounts.

##### **7.2.1.1.2 Language**

**Description:** All text shall be written in English in the game.

**Rationale:** Most potential players of the game understand English and the developers have moderate to extensive knowledge of the language.

**Measure:** A native English speaker should be able to understand how to play the game and the game's objectives.

**Side effects:** Everything has to be written in English.

##### **7.2.1.1.3 Documentation in the game**

**Description:** More extensive documentation about how to play the game shall be included in the first version of the game.

**Rationale:** Players can read the documentation before they start playing a game.

**Unit:** Minutes

**Measure:** A player shall be able to understand the how to play the game and the game's objective by only reading the documentation in a specified number of minutes.



**Goal:** 10 minutes

**Side effects:** The developers have to write documentation about the game.

#### **7.2.1.1.4 Client user interface**

**Description:** The graphical user interface shall be based on XHTML, CSS and JavaScript.

**Rationale:** This type of GUI makes it possible for the player to access the game from different computers as long as the computer has the web browser Mozilla Firefox.

**Unit:** Version of the web browser Mozilla Firefox.

**Measure:** It shall work in the web browser Mozilla Firefox of specified version.

**Goal:** 2.0.0.9

**Side effects:** The developers and the players have to use the specified version of Mozilla Firefox.

#### **7.2.1.2 Performance requirements**

##### **7.2.1.2.1 Client side performance**

**Description:** A computer with the system specification recommended for Mozilla Firefox shall be enough to play the game

**Rationale:** The game is designed for Mozilla Firefox and, therefore, it is likely that the recommended system requirements for Mozilla Firefox are enough to play the game.

**Unit:** RAM in MB, and CPU in Mhz.

**Measure:** The RAM of the computer and the CPU shall be as specified below.

**Goal:** 1000 mb, 1730 Mhz Dual Core.

##### **7.2.1.2.2 Network**

**Description:** Both the server and the client shall be connected to the internet.

**Rationale:** The game is a multiplayer game and, therefore, the server has to be connected to the internet. Hence, the players also need to have internet access to connect with their web browser to the server.

**Unit:** Mbit/s

**Measure:** The client and the server shall be able to access KTH's web site with a web browser to ensure that the computer is connected to internet.

**Goal:** 100 Mbit.

#### **7.2.1.3 Portability requirements**

##### **7.2.1.3.1 Web browsers**

**Description:** The game shall be possible to play with the specified web browser.

**Rationale:** Different web browsers use different standards of the programming languages used. Therefore, it would take too much time implement the game for different web browsers.

**Unit:** Web browser, Version

**Measure:** The player shall be able to play the game with the specified web browser

**Goal:** Mozilla Firefox, 2.0.0.8 or later.

**Side effects:** The developers have to use the specified web browser during the development of the game.

## 7.2.2 Organisational requirements

### 7.2.2.1 Delivery requirements

#### 7.2.2.1.1 Delivery of the requirement document

**Description:** The requirement document shall be delivered on the specified date.

**Rationale:** The developers must have time to complete the design document before the deadline for the final version.

**Unit:** Date

**Measure:** Rand Waltzman shall have the requirement document on the specified date.

**Goal :**January 4, 2008

**Side effects:** The developers have to follow the Gantt chart provided in the appendix of this document.

#### 7.2.2.1.2 Delivery of design document

**Description:** The design document shall be delivered on the specified date.

**Rationale:** The developers must have time to complete the game before the deadline for the final version.

**Unit:** Date

**Measure:** Rand Waltzman shall have the design document on the specified date.

**Goal:** 1 March, 2007 (preliminary date)

**Side effects:** The developers have to make a Gantt chart for the period January to March 2008.

#### 7.2.2.1.3 Delivery of first version

**Description:** The first version of the game shall be provided to Rand Waltzman on the specified date.

**Rationale:** To balance the game correctly the first version will be in beta stage.

**Unit:** Date

**Measure:** Rand Waltzman shall have the first version of the game on the date specified.

**Goal:** 1 May, 2008 (preliminary date)

**Side effects:** The developers have to make a Gantt chart for the period March to May 2008.

### 7.2.2.2 Implementation requirements

#### 7.2.2.2.1 Web server software

**Description:** The game will run on the specified web server.

**Rationale:** There is no time to implement the game on different servers. Besides, the game will not be sold to customer; therefore, it's no need to provide support for more than one server. Apache is open source and, therefore, free to use for the developers.

**Unit:** Web server version

**Measure:** Is shall be possible to install the game on a web server with the specified version of web server software.

**Goal:** Apache HTTP Server (Apache), 2.2 or later.

**Side effects:** During the development of the game the web server need to be the one specified.

#### 7.2.2.2.2 Database management system

**Description:** The game will use the specified database management system (DBMS).

**Rationale:** MySQL is one of the most popular database management systems for web based software. It is also open source and, therefore, free to use for the developers.

**Unit:** DBMS, version

**Measure:** The game shall be able to be in an environment using the specified DBMS.

**Goal:** MySQL, 5.0.32 or later.

**Side effects:** The developers have to use the DBMS specified.

#### 7.2.2.2.3 Server side implementation language

**Description:** The specified programming language shall be used to implement the game on the web server.

**Rationale:** The developers have moderate experience of the programming language. It is used for enterprise web solutions. It is also open source and, therefore, free for the developers to use. PHP is also supported on web hosting providers.

**Unit:** Programming language, version

**Measure:** The game shall be able to run with the specified version of the programming language.

**Goal:** PHP, 5.2.5 or later.

**Side effects:** During the development of the game the specified version of the programming language will be used on the web server.

#### 7.2.2.2.4 Client side implementation languages

**Description:** On the client side of the game Extensible HyperText Markup Language (XHTML), Cascading Style Sheets (CSS) and JavaScript will be used to implement the graphical user interface (GUI) of the game.

**Rationale:** All three languages work with the specified web browser and the developers have moderate knowledge about them.

**Unit:** XHTML version, CSS version, JavaScript version

**Measure:** The specified version of programming languages shall be able to run on Mozilla Firefox and be validated with the World Wide Web Consortium's validators.

**Goal:** 1.1, Level 2, 1.7 or later.

**Side effects:** During the development continues testing shall be done with validators and the web browser Mozilla Firefox specified.

#### 7.2.2.2.5 Version control system

**Description:** During the development of the game a version control system shall be used to keep track of the development.

**Rationale:** There must be a system to keep track of the code during the development; therefore, a version control system is necessary. Even if the developers only have little experience in using version control systems they have to have one. SVN is open source and, therefore, free to use.

**Unit:** Version of version control system

**Goal:** Subversion, 1.4.5 or later.

**Side effects:** The developers have to order a version control system.

### 7.2.2.2.6 Documentation system

**Description:** During the development of the game all documentation shall be saved on the content management system (CMS) specified.

**Rationale:** MediaWiki is easy to use, even for beginners, and it works in different web browsers. It is easy to organize and keep track of different pages in the system. MediaWiki is open source and, therefore, free to use.

**Unit:** Version of CMS

**Goal:** MediaWiki, 1.11.0 or later.

**Side effects:** The developers have to a version of the CMS.

## 7.2.3 External requirements

### 7.2.3.1 Interoperability requirements

#### 7.2.3.1.1 Ethical requirements

##### 7.2.3.1.1.1 Offensive language and pictures

**Description:** No offensive language and pictures in the description of the game, the game's objectives.

**Rationale:** Younger players should be able to play the game. It is outside the scope of the project to develop tools to scan for offensive language and pictures in the communication between players.

**Measure:** A native English speaker should not be able to find any offensive words in the text.

**Side effects:** During the development of the game the person responsible for the project will control that no offensive language is used.

#### 7.2.3.1.2 Privacy requirements

##### 7.2.3.1.2.1 Usage of cookies

**Description:** The game will use cookies to keep players logged in to game. Sweden has laws regulating the use of cookies.

**Rationale:** The game is likely to be hosted on a Swedish web server and, therefore, the game has to follow Swedish regulations of using cookies.

**Measure:** Control that a text is included in the game explaining how cookies are used.

**Side effects:** Create a text explaining how the game uses cookies.

### 7.2.3.2 Safety requirements

#### 7.2.3.2.1 Information saved about the user

**Description:** No personal information about the player will be saved except username, password and e-mail. The web host providers will save information.

**Rationale:** The game is not depended on using personal information; therefore, it is not necessary to ask for them. The players having unique player accounts result in that the game has to save information about username and password.

**Measure:** Control that a text describing what information the game saves about the player is included in the game.

-

**Side effects:** The developers have to write a description of what information that is saved about the user. Ask the web host provider what information they save about the players.

#### **7.2.3.2.2 Secure connection**

**Description:** The game will not provide a secure connection, i.e. encrypt the data transferred between the server and the client. Not even when the players logs in to the game.,

**Rationale:** The developers don't have time to implement the use of secure connection between the server and the client of the game.

**Measure:** Control that a text explaining that the game doesn't provide a secure connection exist.

**Side effects:** The developers have to write a text, which explains that the game doesn't provide secure connection during the log on phase.

## 8 Use Cases

### Table of contents

8 Use Cases.....	38
8.1 Web page.....	39
8.1.1 Create an account.....	39
8.1.2 Log in to an account.....	39
8.2 Game map.....	40
8.2.1 Win a game round.....	40
8.2.2 Start a new round.....	41
8.2.3 Create a game map.....	41
8.2.4 Place players on a game map.....	42
8.3 Resources.....	42
8.3.1 Place resources on a game map.....	42
8.3.2 Gather resources.....	43
8.4 The ship.....	44
8.4.1 Move the ship.....	44
8.4.2 Defeat a player.....	45
8.4.3 Repair the ship.....	45
8.4.4 Move the ship.....	46
8.4.5 Build a module.....	47
8.4.6 Upgrade a module.....	47
8.5 Modules.....	48
8.5.1 Build ammunition (missiles/shells).....	48
8.5.2 Fire missiles.....	48
8.5.3 Fire shells.....	49
8.5.4 Produce main resource.....	49
8.5.5 Teleport the ship.....	50
8.5.6 Hit with a missile.....	50
8.5.7 Hit with a shell.....	51
8.6 High-score lists.....	52
8.6.1 Calculate escape points.....	52
8.7 Alliances.....	52
8.7.1 Create an alliance.....	52
8.7.2 Join an alliance.....	53
8.7.3 Disband an alliance.....	53
8.7.4 Leave alliance.....	54
8.7.5 Dismiss player.....	54
8.8 Communication.....	55
8.8.1 Sending a short text message.....	55
8.9 Research.....	56
8.9.1 Research a research field.....	56

## 8.1 Web page

### 8.1.1 Create an account

**Primary actor:** Player A

**Stakeholders and interests:**

Player A – wants to create an account

**Preconditions:** A uses the specified web browser, has an internet connection and is on the game's web page.

**Minimal guarantee:** The system presents the error.

**Success guarantee:** A creates an account.

**Triggers:** A triggers this process.

**Main success scenario:**

1. A enters the username, password and e-mail address.
2. The system verifies the username, password and e-mail address.
3. The system creates the account.

**Extensions:**

2a. Another player already has the username A specified.

1. The system notifies the user about the already used username.
2. The usecase fails.

2b. The password is invalid.

1. The system notifies the user about the invalid password.
2. The use case fails.

2c. The e-mail address is invalid.

1. The system notifies the user about the invalid e-mail.
2. The use case fails.

### 8.1.2 Log in to an account

**Primary actor:** Player A

**Stakeholders and interests:**

Player A – wants to log in on the game's web page.

**Preconditions:** A is on the game's web page.

**Minimal guarantee:** The system presents the error.

**Success guarantee:** A is logged in to the game.

**Triggers:** A triggers this process.

**Main success scenario:**

1. A enters the username and password.
2. The system verifies the username and password.
3. The system presents information that A is logged in.

**Extensions:**

2a. The username is invalid.

1. The system notifies the user about the invalid username.
2. The use case fails.

2b. The password is invalid.

1. The system notifies the user about the invalid password.
2. The use case fails.

## **8.2 Game map**

### **8.2.1 Win a game round**

**Primary actor:** Player A

**Stakeholders and interests:**

Winning player: wants to win the game round

Other players: wants to win the game round

System: responds to actions

**Preconditions:** Player A is close to the worm hole

**Minimal guarantee:** Nothing happens

**Success guarantee:** Player A wins the game and the round is reset for all players

**Trigger:** Player A triggers this process

**Main success scenario:**

1. Player A moves into the worm-hole in the center of the map.
2. System declares to all players that Player A has won.
3. System awards the top ten players on the escape points ranking list with an addition to their names stating their current rank on the list.
4. System starts a new round.

**Extensions:**

1a: Player A does not have enough energy to move by engines.

1. Use case fails.

1b: Player A does not have enough energy to teleport.

1. Use case fails.



3a: There are less than 10 players playing the game.

1a1: All players in the game is awarded their rank on the escape points ranking list with an addition to their names.

## 8.2.2 Start a new round

**Primary actor:** System

**Stakeholders and interests:**

System: intends to start a new game round.

All players: wants to start playing.

**Precondition:** A player has won a previous game round or a game is started for the first time.

**Minimal guarantee:** A new round is created.

**Success guarantee:** A new round is created.

**Triggers:** System triggers this process.

**Main success scenario:**

1. System creates a new game map.
2. System places all the players on the game map.
3. System starts the new round by giving players control over their accounts.

## 8.2.3 Create a game map

**Primary actor:** System

**Stakeholders and interests:**

System: intends to create a game map

Players: wants the game map to have enough resources and not to be too small or too big.

**Precondition:** System has begun to start a new game round.

**Minimal guarantee:** System notifies of the outcome.

**Success guarantee:** A game map will be created.

**Triggers:** System triggers this process.

**Main success scenario:**

1. System controls the number of players.
2. System calculates the size of the game map in regard to the number of players and a preferred distance between each player.
3. System calculates the amount of recourses on the game map in regard to the number of players.
4. System creates the game map of the right size and places the players on the map.
5. System places the recourses on the game map.

### **Extensions:**

2a There are no players.

1. System notifies of this outcome.
2. Use case fails.

2b There are too many players.

1. System notifies of this outcome.
2. Use case fails.

## **8.2.4 Place players on a game map**

**Primary actor:** System

**Stakeholders and interests:**

System: Intends to place all player ships on the game map.

Players: wants to be placed in an advantageous position in relation to his/hers counterparts.

**Precondition:** System has created a new game map.

System has calculated the amount of players.

**Minimal guarantee:** System notifies of the outcome.

**Success guarantee:** All players ships are placed on the game map.

**Triggers:** System triggers this process.

**Main success scenario:**

1. System calculates the density in which the players ships will be spread in the fringes of the circle formed map based on a preferred distance between each player.
2. System places all the players in a complete circle in the fringes of the circle formed map.

**Extensions:**

2a There are too few players to complete the circle of ships around the map.

1. The system does not make a complete circle but rather places the players ships according to the calculated density.

2b There are too many players and there are players left after the circle is complete.

1. System notifies of the outcome.
2. Use case fails.

## **8.3 Resources**

### **8.3.1 Place recourses on a game map**

**Primary actor:** System

**Stakeholders and interests:**

System: intends to place resources on specific coordinates on the game map.

Players: wants to have enough resources close at hand.

**Precondition:** System has created an empty game map.

System has placed all players on the game map.

System has calculated the amount of resources to be placed.

**Minimal guarantee:** System places the resources on the game map.

**Success guarantee:** System places the resources on the game map.

**Triggers:** System triggers this process.

**Main success scenario:**

1. System calculates the density in which the resources will be spread in different distances from the center (where a closer distance from the center of the map will in general contain a less amount of resources than outer distance) of the map based on preferred resource spread and the current extent of the circle of players (see [Place players on a game map](#)) where if there are no players in a specified perpendicular distance from a straight line going from the center of the map to the fringe of the map, there will be no resources.
2. System divides the resources to smaller amounts that will be represented by resource clouds (clusters of resource squares), and validated that the number of resource clouds comply to the specified density and the size of the calculated area that contains resources on the game map.
3. System randomizes the location of each resource cloud in a circle on each specified distance from the center of the map with a different density at each distance and where players ships can be found within a specified perpendicular distance from a straight line going from the center of the map to the fringe of the map.
4. System calculates the amount of resource squares in each resource cloud.
5. System randomizes the location of each resource square within each resource cloud.

### 8.3.2 Gather resources

**Primary actor:** Player

**Stakeholders and interests:**

Player: wants to gain resources.

System: give the right amount of resources.

Precondition: Player has an account.

**Minimal guarantee:** Player will be notified of the outcome.

**Success guarantee:** Player will gain resources.

**Triggers:** Player triggers this process.

**Main success scenario:**

1. Player triggers the collect secondary-resource function.
2. System validates that the square contains secondary-resources.

3. System deducts an amount of the main resource from Players main resource storage module.
4. System deducts an amount of secondary-resources from the resource square.
5. System adds an amount of secondary-resources to Players secondary-resource module.

**Extensions:**

2a There are no resources in the resource square.

1. Use case fails.

3a There is not enough of the main resource in Players main resource storage module.

1. Player is notified of the outcome.
2. Use case fails.

4a There is a less amount of resources in the resource square than the System tries to deduct.

1. System deducts the rest of the resources from the square.

5a There is not enough space left in Players resource modules for the new addition of the secondary-resource.

1. System fills up the storage.
2. System discards the rest of the resources.

## **8.4 The ship**

### **8.4.1 Move the ship**

**Primary actor:** Player

**Stakeholders and interests:**

Player: wants to move the ship.

System: responds to events.

**Precondition:** Player has an account.

**Minimal guarantee:** Player will be notified of the outcome.

**Success guarantee:** Players moves the ship.

**Trigger:** Player triggers this process.

**Main success scenario:**

1. Player triggers the move ship function.
2. System deducts an amount of resources depending on the amount of engine research done by Player, from Players resource storage modules.
3. System initiates a countdown that indicates when the ship movement is done. Time it takes to travel depends on the amount of engines modules and the engine modules level.
4. When the countdown is finished, the ship is in the designated square and stops.

**Extensions:**

2a Player lack the required amount of resources.

1. Player is notified of the outcome.
2. Use case fails.

4a Two ships are in the same map square.

1. A collision with the other ship occurs.
2. Players ship stops.
3. Countdown stops.
4. Use case fails.

### 8.4.2 Defeat a player

**Primary actor:** Player

**Stakeholders and interests:**

Attacker: wants to defeat Player

Player: wants to survive.

System: Calculate all events.

**Precondition:** Attacker has reduced Players ships condition status to zero or below.

**Minimal guarantee:** Player restarts.

**Success guarantee:** Player restarts.

**Trigger:** System triggers this process.

**Main success scenario:**

1. Player is notified that his/her ship is destroyed and that he/she is defeated.
2. System created a new default ship and places it in the fringe of the map.
3. System generates a small resource cloud around the Players new ship.
4. System gives over the control of the account to Player.

### 8.4.3 Repair the ship

**Primary actor:** System

**Stakeholders and interests:**

Player: wants the ship repaired.

System: responds to events.

**Precondition:** Player has a repair module.

**Minimal guarantee:** Nothing happens.

**Success guarantee:** Players ships ship condition is improved.

**Trigger:** System triggers this process.

### **Main success scenario:**

1. System validates that the ships condition status is less than 100%
2. System deducts an amount of resources depending on the amount of repair research done by Player from Players resource storage modules.
3. System initiates a countdown that indicates when the repair work is done. Time it takes to repair depends on the amount of repair modules and the repair modules level.
4. When the countdown is finished, the ships condition status is increased.

### **Extensions:**

2a Ships condition status is 100%

1. Use case fails.

3a Player lack the required amount of resources.

1. Use case fails.

5a Players ship is destroyed before countdown reaches zero.

1. Player is defeated

2. Use case fails.

## **8.4.4 Move the ship**

**Primary actor:** Player

**Stakeholders and interests:**

Player: wants to move the ship.

System: responds to events.

**Precondition:** Player has an account.

**Minimal guarantee:** Player will be notified of the outcome.

**Success guarantee:** Players moves the ship.

**Trigger:** Player triggers this process.

### **Main success scenario:**

1. Player triggers the move ship function.
2. System deducts an amount of resources depending on the amount of engine research done by Player, from Players resource storage modules.
3. System initiates a countdown that indicates when the ship movement is done. Time it takes to travel depends on the amount of engines modules and the engine modules level.
4. When the countdown is finished, the ship is in the designated square and stops.

### **Extensions:**

2a Player lack the required amount of resources.

1. Player is notified of the outcome.

2. Use case fails.
- 4a Two ships are in the same map square.
1. A collision with the other ship occurs.
  2. Players ship stops.
  3. Countdown stops.
  4. Use case fails.

### 8.4.5 Build a module

**Primary Actor:** System

**Stakeholders and interest:**

Stakeholder1: System – Build a module

Stakeholder2: Player – Build a module

**Precondition:** Player has an account.

**Minimal guarantee:** System sends information about the outcome

**Success guarantee:** A module is built

**Triggers:** The player triggers this process

**Main Success Scenario:**

1. Player indicates to the system that he/she desires to build a specific module
2. System receives information and deduct the required resources, for this module, from the player
3. System produces the desired module

**Extensions:**

2.a System cannot build a specific module due to an insufficient amount of resources

1. System sends information about the outcome
2. Use Case fails

2.b System cannot build a specific module because Player not made adequate research for this module

1. System sends information about the outcome
2. Use Case fails

### 8.4.6 Upgrade a module

**Primary Actor:** System

**Stakeholders and interest:**

Stakeholder1: System – Upgrade a module

Stakeholder2: Player – Upgrade a module

**Precondition:** The player has built the module he/she wishes to upgrade

**Minimal guarantee:** System sends information about the outcome

**Success guarantee:** A module is upgraded

**Triggers:** The player triggers this process

**Main Success Scenario:**

1. Player indicates to the system that he/she desires to upgrade a specific module
2. System receives information and deduct the required resources, for upgrading this module, from the player
3. System upgrades the desired module.

**Extensions:**

2.a System cannot upgrade a specific module due to an insufficient amount of resources.

1. System sends information about the outcome.
2. Use Case fails.

## **8.5 Modules**

### **8.5.1 Build ammunition (missiles/shells)**

**Primary Actor:** System

**Stakeholders and interest:**

Stakeholder1: Player – Acquire ammunition

Stakeholder2: System – Produce ammunition

**Precondition:** The player has a weapons module.

**Minimal guarantee:** System sends information about the outcome

**Success guarantee:** Ammunition is built

**Triggers:** The player triggers this process

**Main Success Scenario:**

1. Player indicates to the system that he/she desires to build ammunition
2. System deduct the required resources for building ammunition
3. System produces the desired ammunition

**Extensions:**

2.a System cannot build ammunition due to an insufficient amount of resources.

1. System sends information about the outcome.
2. Use Case fails.



## 8.5.2 Fire missiles

**Primary Actor:** Player

**Stakeholders and interest:**

Stakeholder1: Player – Fire missiles

Stakeholder2: System – Fire missiles

**Precondition:** The player has a missile module and has missiles stored.

**Minimal guarantee:** A missile is fired

**Success guarantee:** A missile is fired

**Triggers:** Player triggers this process

**Main Success Scenario:**

1. Player indicates to the system that he/she desires to fire a missile at a specific ship
2. System receives information and calculates the impact time based on the given coordinates for the other ship relative to the player's ship and the level of missile research.
3. System notifies the player that the missiles have been fired and the calculated impact time.
4. System starts counting down the impact time.
5. System notifies the player of the outcome.

## 8.5.3 Fire shells

**Primary Actor:** Player

**Stakeholders and interest:**

Stakeholder1: Player – Fire shells

Stakeholder2: System – Fire shells

**Precondition:** The player has a canon module and shells for that module.

**Minimal guarantee:** The system notifies the player of the outcome

**Success guarantee:** A shell is fired

**Triggers:** The player triggers this process.

**Main Success Scenario:**

1. Player indicates to the system that he/she desires to fire a shell at a specific ship
2. System receives information and calculates the impact time based on the given coordinates for the other ship relative to the player's ship and the level the cannons have been researched
3. System notifies the player that the shells have been fired and the calculated impact time.
4. System starts counting down the impact time.

**Extensions:**

3.a The range of the cannon is sufficient

1. System notifies the player that the selected target is out of the cannons range
2. Use Case fails

#### **8.5.4 Produce main resource**

**Primary actor:** System

**Stakeholders and interests:**

System – deliver the main resource.

Player – receive the daily income of the main resource to have at disposal.

**Precondition:** Player needs at least one power plant to receive the main resource.

**Minimal guarantee:** System notifies Player of the outcome

**Success guarantee:** System delivers an amount of the main resource to Player

**Triggers:** System triggers this process.

**Main success scenario:**

1. System validates how many power plants Player have.
2. System validates the level of the Player's power plants.
3. System calculates the amount of the main resource that will be delivered to Player .
4. System validates that the amount of the main resource to be delivered can be stored on Player's ship.
5. System delivers an amount of the main resource to Player.

**Extensions:**

4a Some of the amount of resource to be delivered can't be stored.

1. The resources that could not be stored will not be given to A.
2. Use-Case fails

4b None of the amount of resource to be delivered can be stored.

1. None of the resources that would've been delivered to A will be given to A.
2. Use-Case fails

#### **8.5.5 Teleport the ship**

**Primary actor:** Player

**Stakeholders and interests:**

Player – teleport the ship.

Precondition: Player needs to have the teleport-module on the ship.

**Minimal guarantee:** System notifies the player of the outcome

**Success guarantee:** Player's ship have been teleported

**Triggers:** Player triggers this process.

**Main success scenario:**

1. Player choose to teleport
2. Player's ship gets teleported
3. Player's ship ends up on a random location close to where the ship previously were.

**Extensions:**

1a Not enough resources.

1. User is notified of the outcome
2. Use case fails.

3a Attacks were incoming towards the ship.

1. All attacks that were incoming towards the ship are canceled.

### **8.5.6 Hit with a missile**

**Primary actor:** System

**Stakeholders and interests:**

System – respond to user demands

Player A– damage Player B with a missile.

Player B– avoid incoming missile from A.

**Precondition:** A has launched a missile towards B, the missile now travels towards B.

**Minimal guarantee:** A and B will be notified of the outcome.

**Success guarantee:** A hit B with a missile.

**Triggers:** Player triggers this process.

**Main success scenario**

1. System counts down the impact time of the missile
2. After the count down is done, System validates A's research level in missiles
3. System validates B's level of missile decoys and missile decoy research.
4. The missile hit, system applies damage to B.

**Extensions:**

4a B teleports.

1. Use case fails.

4b B's missile decoys intercepts the missile.

1. Use case fails.

4c B's ships condition status reaches 0 or below.

1. B's ship is destroyed.

## 8.5.7 Hit with a shell

**Primary actor:** System

**Stakeholders and interests:**

System – determine if the shell hit its target and how much damage it does.

Player A– wants to damage B with a shell.

Player B– wants to avoid the shell.

**Precondition:** A has launched a shell towards B, the shell now travels towards B.

**Minimal guarantee:** A and B will be notified of the outcome.

**Success guarantee:** A hit B with a shell.

**Triggers:** System triggers this process.

**Main success scenario:**

1. System counts down the calculated impact time for the shells
2. System validates that the shells traveling time has reached zero.
3. The shell hit, system applies damage to B.

**Extensions:**

3a B teleports.

1. Use case fails.

3b B's ships condition status reaches 0 or below.

1. B's ship is destroyed.

## 8.6 High-score lists

### 8.6.1 Calculate escape points

**Primary actor:** System

**Stakeholders and interests:**

System – want to compute the list of escape points

Precondition: A game is running

**Minimal guarantee:** System notifies game-administrator about eventual failure

**Success guarantee:** The list is created

**Triggers:** System timer triggers the process.

**Main success scenario:**

1. The system timer notifies the system to recalculate the escape points list.
2. The system validates needed information for the calculation.
3. The system calculates the escape points list.

4. The system updates the list with the newly calculated escape points

## **8.7 Alliances**

### **8.7.1 Create an alliance**

**Primary actor:** Player A

**Stakeholders and interests:**

Player A – wants to create an alliance.

**Preconditions:** A is logged in on the game's web page.

**Minimal guarantee:** The system presents the outcome.

**Success guarantee:** A creates an alliance.

**Triggers:** A triggers this process.

**Main success scenario:**

1. A enters name of the alliance.
2. The system verifies the name of the alliance.
3. The system creates the alliance.

**Extensions:**

1a. The player is already member of an alliance.

1. Use case fails.

2a. The name of the alliance already exists.

1. The system notifies the user about the already used alliance name.
2. The use case fails.

### **8.7.2 Join an alliance**

**Primary actor:** Player A

**Stakeholders and interests:**

Player A – wants to join an alliance.

Player B – wants A to join the alliance.

**Preconditions:** B is the alliance leader of the alliance.

**Minimal guarantee:** The system presents the outcome

**Success guarantee:** A joins an alliance.

**Triggers:** A triggers this process.

**Main success scenario:**

1. A enters the name of the alliance to join.
2. B accepts A to join.

3. The system presents information for all other players in the alliance that A has joined.

**Extensions:**

1a. The name of the alliance does not exist.

1. The system presents the error.
2. The use case fails

2a. B does not accept A.

1. Use case fails.

### 8.7.3 Disband an alliance

**Primary actor:** Player A

**Stakeholders and interests:**

Player A – wants to disband the alliance.

Other players in the alliance - also wants the alliance to be disband.

**Preconditions:** A is the alliance leader of the alliance.

**Minimal guarantee:** The system presents the outcome.

**Success guarantee:** The alliance is disbanded.

**Triggers:** A triggers this process.

**Main success scenario:**

1. A decides to disband the alliance.
2. The system presents information for all other players in the alliance that the alliance will be disbanded in 24 hours.
3. The system disbands the alliance.

**Extensions:**

2a. Another player in the alliance takes over as alliance leader.

1. The alliance will not be disbanded.
2. The system will assign the player to be new alliance leader.

2b. A cancels the disbanding of the alliance.

1. The alliance will not be disbanded.
2. A will continue as alliance leader.

### 8.7.4 Leave alliance

**Primary actor:** Player A

**Stakeholders and interests:**

Player A – wants to leave the alliance.

**Preconditions:** A is a member of an alliance.

**Minimal guarantee:** The system presents the outcome.

**Success guarantee:** A leaves the alliance.

**Triggers:** A triggers this process.

**Main success scenario:**

1. A decides to leave the alliance.
2. The system presents information for all other players that A leaves the alliance in 24 hours.

**Extensions:**

2a. A cancels the leaving from the alliance.

1. A is still a member of the alliance.

### **8.7.5 Dismiss player**

**Primary actor:** Player A

**Stakeholders and interests:**

Player A – wants B to be dismissed from the alliance.

Player B – wants to stay as member of the alliance.

**Preconditions:** A and B is member of the same alliance. A is the alliance leader of the alliance.

**Minimal guarantee:** The system presents the outcome.

**Success guarantee:** B is dismissed from the alliance.

**Triggers:** A triggers this process.

**Main success scenario:**

1. A decides that B shall be dismissed from the alliance.
2. The system presents information for B that he will be dismissed in 12 hours.
3. The system dismisses B from the alliance and presents information for all other players in the alliance that B is dismissed from the alliance.

**Extensions:**

2a. A decides not to dismiss B.

1. A will continue to be a member of the alliance.

2b. B decides to leave the alliance.

1. B leaves the alliance.

## **8.8 Communication**

### **8.8.1 Sending a short text message**

**Primary actor:** Player A

**Stakeholders and interests:**

Player A – want to send a message to Player B

Player B – want the ability to receive messages

System – wants to deliver message without errors

**Precondition:** Player A has an active game-account

**Minimal guarantee:** Player A gets notified if the message was delivered

**Success guarantee:** Player B receives the message

**Triggers:** Player A triggers the process.

**Main success scenario:**

1. Player A notifies the system that he want to send a message M to Player B.
2. The system to delivers the message M to Player B.
  1. The system notifies Player A that the message has been delivered.

**Extensions:**

1a The Player B does not exist

1. The system notifies player A that Player B does not exist
2. The use case fails.

2a The system could not deliver the message

1. The system notifies player A that Player B did not receive the message.
2. The use case fails

## **8.9 Research**

### **8.9.1 Research a research field**

**Primary actor:** Player A

**Stakeholders and interests:**

Player A – wants to research

System – responds to events

**Precondition:** Player A has an active account

**Minimal guarantee:** Player A is notified of the outcome

**Success guarantee:** The research is carried through

**Triggers:** Player A triggers the process.

**Main success scenario:**

1. Player A notifies the system that he want to research in a research field.
2. The system calculates the time left for the research to be done and notifies Player A.
3. The system validates when the research time has reached zero and completes the research.



**Extensions:**

2a Player A does not have enough resources for the research.

1. The system notifies Player A that he has too little resources to carry through the research.
2. Use case fails.

## **9 System evolution**

### ***9.1 Fundamental assumptions on which the system is based:***

#### **9.1.1 Client**

- The player shall operate in a MS Windows environment.
- The player shall operate in the web browser Mozilla Firefox.
- The player shall have internet access.

#### **9.1.2 Server**

- The server shall use open-source solutions for web server and database.

### ***9.2 Anticipated future changes to the system***

The development of new versions for the web browser Mozilla Firefox may have an effect on the system.

New versions of the server software, for example the web server and database, may have an effect on the system.

User requirements for the system are likely to change in the future to balance the game's difficulty.

## **10 Appendices**

### ***10.1 Database descriptions***

The database will be used to store information about the users, including

- their ships condition status
- their resources
- their user names, passwords and emails
- their ship modules and upgrades
- their position on the map
- their researches
- their escape points and distances towards the worm-hole
- their message-history
- their alliances

How the database tables will be constructed and organized has not yet been decided.

## 11 Alphabetical Index

- 3D graphics 6
- AI 6
- AJAX 7
- alliances 15
- Alliances
  - Create an alliance 53
  - Disband an alliance 54
  - Dismiss player 55
  - Join an alliance 53
  - Leave alliance 54
  - Players in an alliance 30
- Animation 6
- Apache HTTP Server 34
- appeal 7
- Calculate escape points 52
- Chat 6
- Client user interface 33
- CMS 36
- Communication 15, 30
- cookies 36
- CSS 8, 33, 35
- Database management system 35
- DBMS 35
- Delivery 17
- Delivery requirements 17, 34
- Documentation 32
- Documentation system 36
- Escape points 29
- Ethical requirements 18, 36
- External requirements 18, 36
- forum 30
- Functional requirement 12
  - 5.1.2 Game map 13
    - Create a game map 41
    - Game map 22
    - Main resource 23
    - Place players on a game map 42
    - Player view 23
    - Resource squares 23
    - Secondary resource 23
    - Start a new round 41
    - Startup placement of players 23
    - The wormhole 22
    - Win a game round 40
- High-score 29
- HTML 8
- Implementation 17
- Implementation requirements 17, 34
- Information saved about the user 36
- Interoperability requirements 18, 36
- Intuitive 7
- Javascript 7
- JavaScript 33, 35
- Language 32
- Login 32
- massive multi player 5
- MediaWiki 36
- messages 15
- Modules 14
  - Build ammunition (missiles/shells) 48
  - Building decoys 28
  - Building missiles 26
  - Building shells 26
  - Cannons module 26
  - Canons 14
  - Capacity of decoys 27
  - Defensive Weapons 27
  - engine 14
  - Engine module 29
  - Escape points 15
  - Fire missiles 49
  - Fire shells 49
  - Firing cannons 27
  - Hit with a missile 51
  - Hit with a shell 52
  - Launch missiles 26
  - Missile batteries 14
  - Missile batteries module 25
  - missile decoy 14
  - Missile decoys module 27
  - Nullifying missile attack 28
  - Offensive Weapons 25
  - Power plant 15
  - Power Plant 29
  - Produce main resource 50
  - repair 15
  - Repair module 29
  - Storage 14
  - Storage module 28
  - Storage of missiles 25

- Storage of shells 26
- Teleport the ship 50
- teleportation 14
- Teleportation module 27
- wormhole 15
- move 15
- Mozilla Firefox 6, 33, 35
- Music 6
- MySQL 8, 35
- Non-functional requirements 16, 32
- Offensive language 36
- Organisational 17
- Organisational requirement 17
- Organisational requirements 34
- PC 5
- Performance 16
- Performance requirements 16, 33
- PHP 8, 35
- Portability 17
- Portability requirements 17, 33
- Privacy requirements 18, 36
- Product requirements 16, 32
- Rand Waltzman 34
- research 15
  - cannons 31
  - Cannons 16
  - engines 31
  - Engines 16
  - missile decoys 31
  - Missile decoys 16
  - missiles 30
  - Missiles 15
  - repair 32
  - Repair 16
  - Research a research field 56
  - teleportation 31
  - Teleportation 16
- resource 5, 13
- resources
  - Gather resources 43
  - Place recourses on a game map 42
- Resources 13
- role-playing 5
- Safety requirements 18, 36
- scope
  - Inside 6
  - Outside 6
- Secure connection 37
- ship
  - Build a module 47
  - Damage ship 24
  - Default ship 24
  - Defeat a player 45
  - Destroying ship 24
  - Move the ship 44, 46
  - Movements of ship 25
  - Repair ship 24
  - Repair the ship 45
  - Upgrade a module 47
- Skills star 29
- social sphere 5
- Sound Effects 6
- SQL 8
- Stability 7
- SVN 35
- system architecture 19
- Technologies 7
- Text messages 30
- The ship 13
- Usability 6, 16
- Usability requirements 16, 32
- User requirements 12
  - 5.1.1 Web page 12
    - Create an account 21, 39
    - Forum 22
    - Log in 22
    - Log in to an account 39
- Web server 34
- Version control system 35
- World Wide Web Consortium 35
- wormhole 13
- XHTML 33, 35
- version of the CMS. 36